Network Bound Disk Encryption

Vincent Brobald
Senior Consultant - Devoteam

Disk Encryption – Why ?

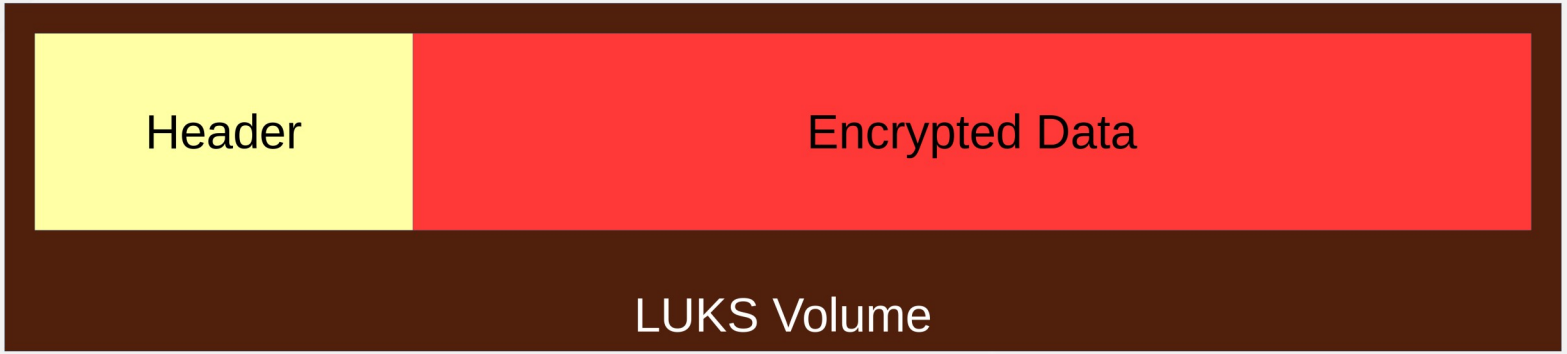
- Secure data against device theft or loss (laptop, hdd, usb key)
- Secure data against weak SAN transport
- Secure data against storage provider
- Prevent system tampering
- Ensure implicit disk sanitizing

Disk Encryption – How ?

Linux Unified Key Setup

LUKS is the standard for GNU/Linux hard disk encryption. By providing a standard on-disk-format, it does not only facilitate compatibility among distributions, but also provides secure management of multiple user passwords.

Anatomy of Luks



LUKS stores all necessary setup information in the partition header, enabling to transport or migrate data seamlessly.

Anatomy of Luks - header

```
LUKS header information for /dev/sda2

Version:        1
Cipher name:    aes
Cipher mode:    xts-plain64
Hash spec:      sha256
Payload offset: 4096
MK bits:        512
MK digest:      75 f6 89 6e d0 18 9f 0c 6f 3e be cc 21 7a 90 74 1c 48 44 d1
MK salt:        ac f9 41 c1 a9 42 67 f1 a5 31 92 c4 8a 81 e3 5f
                1b 51 27 a4 82 b2 dc 8e a6 b5 ef 80 ae 11 fc 2a
MK iterations:  89775
UUID:           df369dc4-ccbe-4249-99ae-f20dd9124cfb

Key Slot 0: ENABLED
        Iterations:             1436404
        Salt:                   1b ff a5 2a 10 32 88 5d 40 9d c4 fe 1c 65 ea 82
                                2b c3 e4 6a 5e f9 35 c7 b7 76 01 26 64 e9 49 9c
        Key material offset:    8
        AF stripes:             4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

Disk Encryption – What ?

Local storage

Remote Storage (SAN)

Cloud storage

Mobile storage (USB keys, USB HDD, ...)

Escrow – The “Setec Astronomy” problem.

Traditional Key Escrowing, AKA vault

- Stores centrally all secrets
- Heavy infrastructure requirements
- Heavy security requirements (Authentication, RBAC, ...)
- Secrets exists in multiple places
- Replication required for HA

Escrow – The “Too Many Secrets” problem (2)

Escrow process, by definition, transmit sensitive secrets on the network, therefore putting the secret at risk in case of protocol weakness.

The recent years have proved stressful for technologies like SSL/TLS (f.e. Heartbleed)

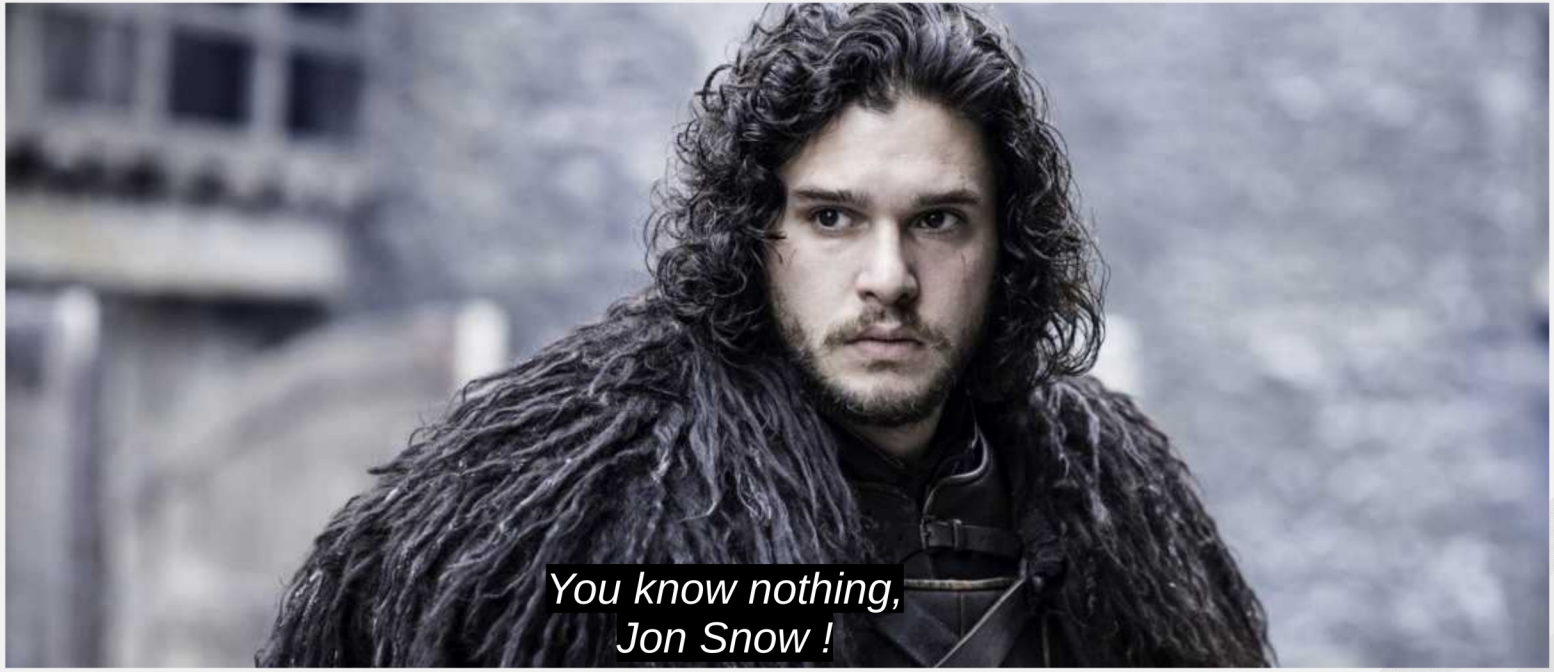
The best way to ensure your secret is kept secret is to keep it to yourself.

Tang

Network-Based Cryptographic Binding Server

- Tang is not a key escrow solution
- Tang does not want to know your secrets
- Tang is stateless
- Tang doesn’t need encrypted transport
- Tang doesn’t know or care what it is talking with
- Tang does not care about high availability
- Tang knows almost nothing

Tang is a bit like this guy ...



Ok, but what is it ?

Tang is a lightweight rest server that is used by local client to unlock an intermediate secret used to decypher the disk encryption key.

It just gets a challenge from the client, sends a cryptographic response and the response is used to rebuild the secret to decrypt the disk encryption key.

Meet Clevis

- Clevis is the client that performs volume enrollment and unlocking.
- Clevis can exist in multiple system contexts :
  - When the system boots (dracut for unlocking your root partition)
  - When the server processes fstab (integrated with systemd)
  - When you insert a new volume (udisk2 integration)

Meet Clevis (2)

Clevis can work with different ‘pin’ providers :

- TPM2
- Single Tang
- Multiple Tang with Shamir Secret Sharing

Clevis and Tang – let's play

```
# echo hi | clevis encrypt tang '{"url": "http://tang1.tangdemo.lab"}' > hi.jwe
The advertisement contains the following signing keys:

EoDpPeCkeKVDsqOlTZuyudu4PEA

Do you wish to trust these keys? [ynYN] y
# cat hi.jwe
EyJhbGciOiJFQ0RILUVTIiwiY2xldmlzIjp7InBpbiI6InRhbmciLCJ0YW5nIjp7ImFkdiI6eyJrZXlzIjpbeyJhbGciOiJFUzUxMiIsImNydiI6IlAtNTIxIiwia2V5X29wcyI6WyJ2ZXJpZnkiXSwia3R5IjoiRUMiLCJ4IjoiQUw5eEZDUFBWa1h1czNzdTdpVGJvWi1Qdmd1VkRpOThlMm4ySGtwSnRPRFhSQUI1X3B1TFhHQ2VRSnJZQmp6S3UzRVgzVTFUWFQ5SUJsdk1id29sUktVNSIsInkiOiJBR…
# clevis decrypt < hi.jwe
hi
```

Clevis and Tang – binding a volume

```
[root@client1 ~]# clevis luks bind -d /dev/vda2 tang '{"url": "http://tang1.tangdemo.lab"}'
The advertisement contains the following signing keys:

EoDpPeCkeKVDsqOlTZuyudu4PEA

Do you wish to trust these keys? [ynYN] y
You are about to initialize a LUKS device for metadata storage.
Attempting to initialize it may result in data loss if data was
already written into the LUKS header gap in a different format.
A backup is advised before initialization is performed.

Do you wish to initialize /dev/vda2? [yn] y
Enter existing LUKS password:
```

Clevis and Tang - Enrollment

- Celvis generate a key based on the disk encryption key entropy
- Clevis negotiates the creation of an intermediate secret with Tang
- Clevis derives the disk encryption key so it is only recoverable with the secret that only the right Tang instance can help generate.
- Clevis stores the data necessary to replay the negotiation with Tang in a LUKS key slot.

SSS for HA and

High Availability in Clevis/Tang is achieved using Shamir Secret Share. SSS is a secret sharing algorithm allowing to divide a secret between multiple parties with a defined threshold allowing to rebuild the secret using any group of parties as long as it reaches the threshold.

High availability: 2 tang servers in sss with a threshold of one server

High availability with multi-tang validation: 4 tang servers with a threshold of 3

Other security concerns

- Consider rotating your keys in Tang, but be careful with cold-stored encrypted disks
- In IAAS, do not forget that if you start deploying your servers from the same encrypted image, they will end-up with the same encryption key; consider deploying secondary disks initialized individually with LUKS for sensitive data.
- Do not install Tang servers under the same technical authority as your clients. Make Tang servers available to your infrastructure through VPN.

Questions ?

Selected questions from Q&A segment:

Q: What happens if all Tang servers are down (f.ex due to an attack)?

A: Active systems will not be impacted as long as they do not try to unlock a managed volume. Systems booting will wait until Tang infrastructure is available again.

Q: What is the impact for backups?

A: File-based backups are not impacted. Block-based backups will work as well. Tang key rotation may however have an impact on the feasibility to recover files from an old backup. Backup systems that perform block-based backup and file restore need to support clevis and have access to tang server.

Q: how does the root volume unlocks at boot?

A: if initramfs has been properly regenerated with dracut, the boot process will start the network before the root volume is unlocked, allowing the clevis-dracut integration to query the tang server.